Slides: speakerdeck.com/rmoff/
Twitter: @rmoff

Enabling Rapid Business Insight into Fast Data with Oracle Stream Analytics and Oracle GoldenGate

OUGN 2017

Robin Moffatt, Rittman Mead

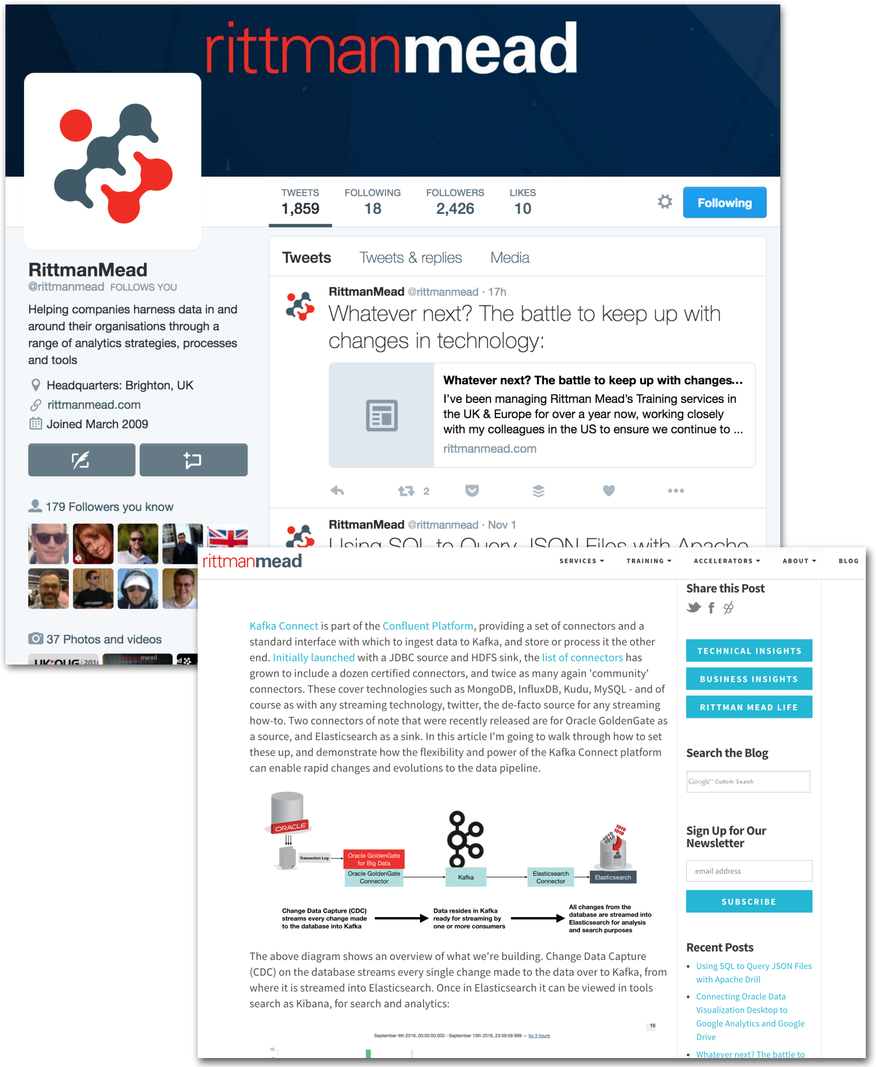
Robin Moffatt

- Head of R&D, Rittman Mead
  - Previously OBIEE/DW developer at large UK retailer
  - Previously SQL Server DBA, Business Objects, DB2, COBOL….
- Oracle ACE
- Frequent blogger : http://ritt.md/rmoff and http://rmoff.net
- Twitter: @rmoff
- IRC: rmoff / #obihackers / freenode

Rittman Mead - OBIEE System metrics
Analytics with Kibana and Elasticsearch through Hadoop – part 3 – Visualising the data in Kibana
Analysing ODI performance with Flame
Graph Analysis of the Panama Papers with Oracle Big Data Spatial and Graph
Officer
Intermediary
Entity
officer_of
intermediary_of
similar

Rittman Mead

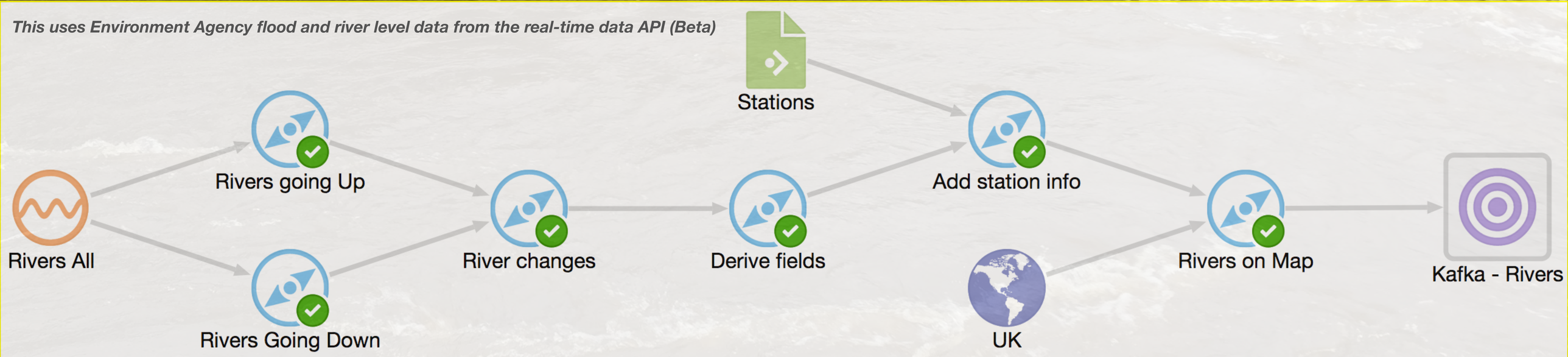
- Oracle Gold Partner with offices in the UK and USA
- 70+ staff delivering Oracle BI, DW, Big Data and Advanced Analytics projects
- Significant web presence with the Rittman Mead Blog (http://www.rittmanmead.com)
- Hadoop R&D lab for “dogfooding” solutions developed for customers


rittmanmead
TWEETS 1,859
FOLLOWING 18
FOLLOWERS 2,426
LIKES 10
Following
RittmanMead
@rittmanmead FOLLOWS YOU
Helping companies harness data in and around their organisations through a range of analytics strategies, processes and tools
Headquarters: Brighton, UK
rittmanmead.com
Joined March 2009
179 Followers you know
37 Photos and videos
Tweets
Tweets & replies
Media
RittmanMead @rittmanmead · 17h
Whatever next? The battle to keep up with changes in technology:
Whatever next? The battle to keep up with changes...
I've been managing Rittman Mead's Training services in the UK & Europe for over a year now, working closely with my colleagues in the US to ensure we continue to ...
rittmanmead.com
RittmanMead @rittmanmead · Nov 1
rittmanmead
SERVICES
TRAINING
ACCELERATORS
ABOUT
BLOG
Kafka Connect is part of the Confluent Platform, providing a set of connectors and a standard interface with which to ingest data to Kafka, and store or process it the other end. Initially launched with a JDBC source and HDFS sink, the list of connectors has grown to include a dozen certified connectors, and twice as many again 'community' connectors. These cover technologies such as MongoDB, InfluxDB, Kudu, MySQL - and of course as with any streaming technology, twitter, the de-facto source for any streaming how-to. Two connectors of note that were recently released are for Oracle GoldenGate as a source, and Elasticsearch as a sink. In this article I'm going to walk through how to set these up, and demonstrate how the flexibility and power of the Kafka Connect platform can enable rapid changes and evolutions to the data pipeline.
ORACLE
Transaction Log
Oracle GoldenGate for Big Data
Oracle GoldenGate Connector
Kafka
Elasticsearch Connector
Elasticsearch
Change Data Capture (CDC) streams every change made to the database into Kafka
Data resides in Kafka ready for streaming by one or more consumers
All changes from the database are streamed into Elasticsearch for analysis and search purposes
The above diagram shows an overview of what we're building. Change Data Capture (CDC) on the database streams every single change made to the data over to Kafka, from where it is streamed into Elasticsearch. Once in Elasticsearch it can be viewed in tools search as Kibana, for search and analytics:
Share this Post
TECHNICAL INSIGHTS
BUSINESS INSIGHTS
RITTMAN MEAD LIFE
Search the Blog
Sign Up for Our Newsletter
email address
SUBSCRIBE
Recent Posts
Using SQL to Query JSON Files with Apache Drill
Connecting Oracle Data Visualization Desktop to Google Analytics and Google Drive
Whatever next? The battle to

“Stream” Processing…

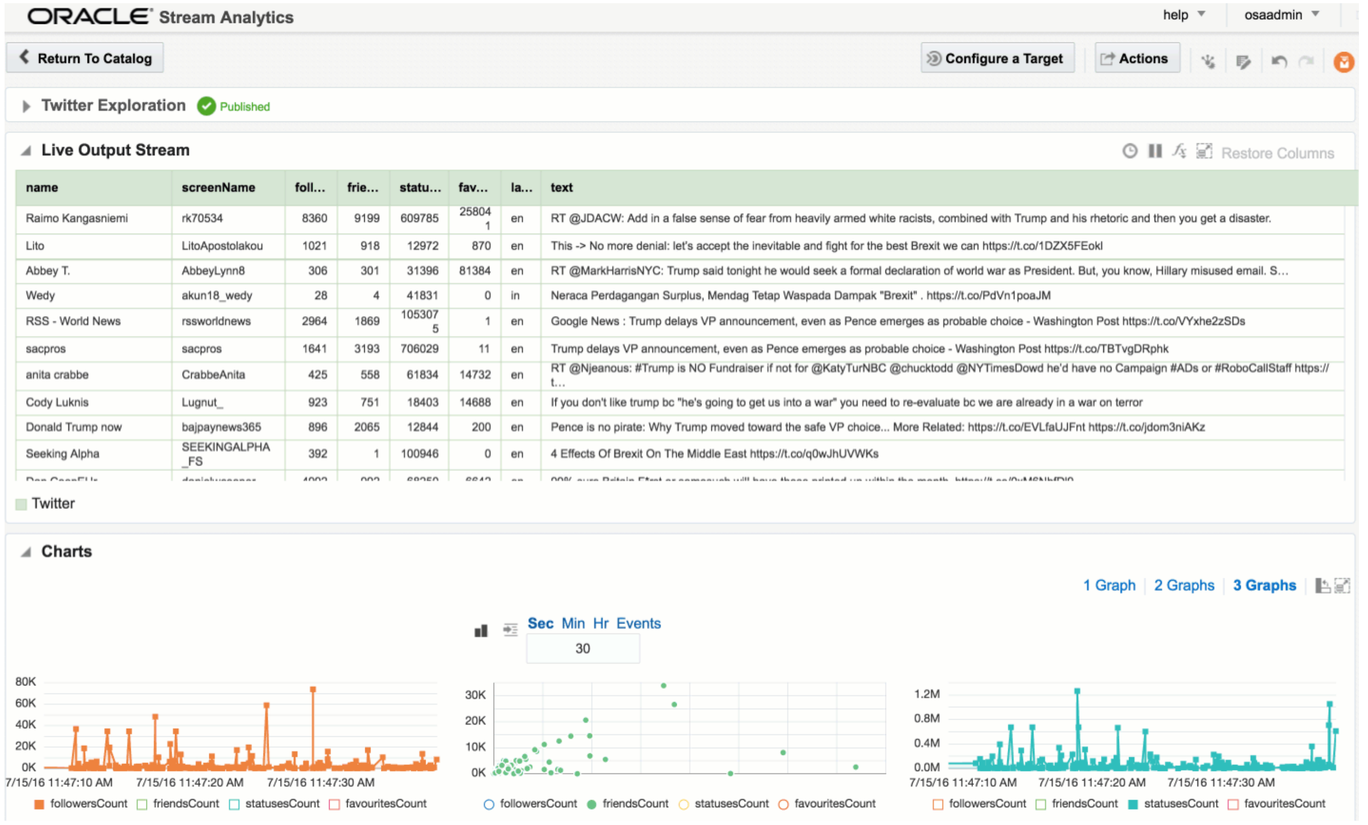
“Stream” Processing…

This uses Environment Agency flood and river level data from the real-time data API (Beta)

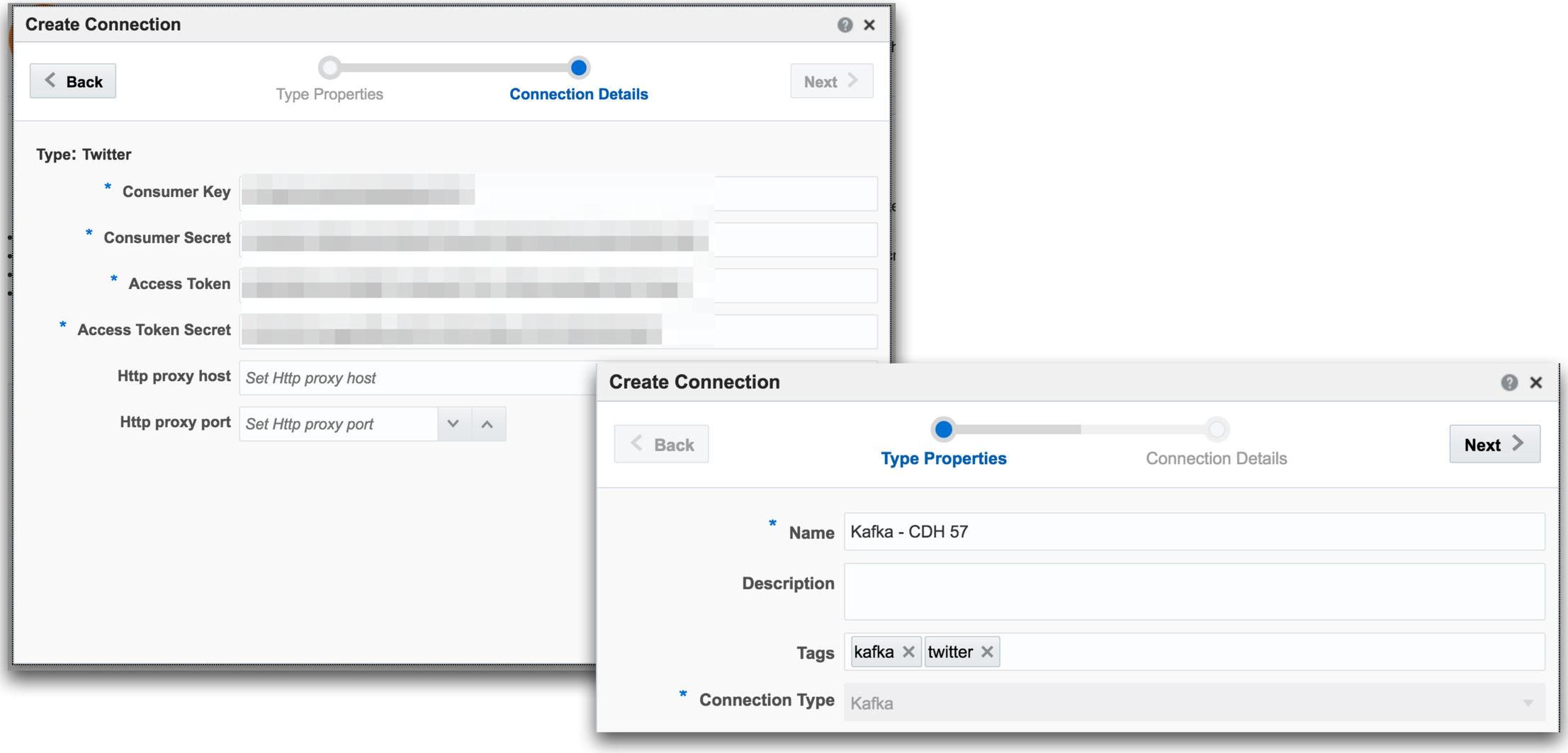


Oracle Stream Analytics

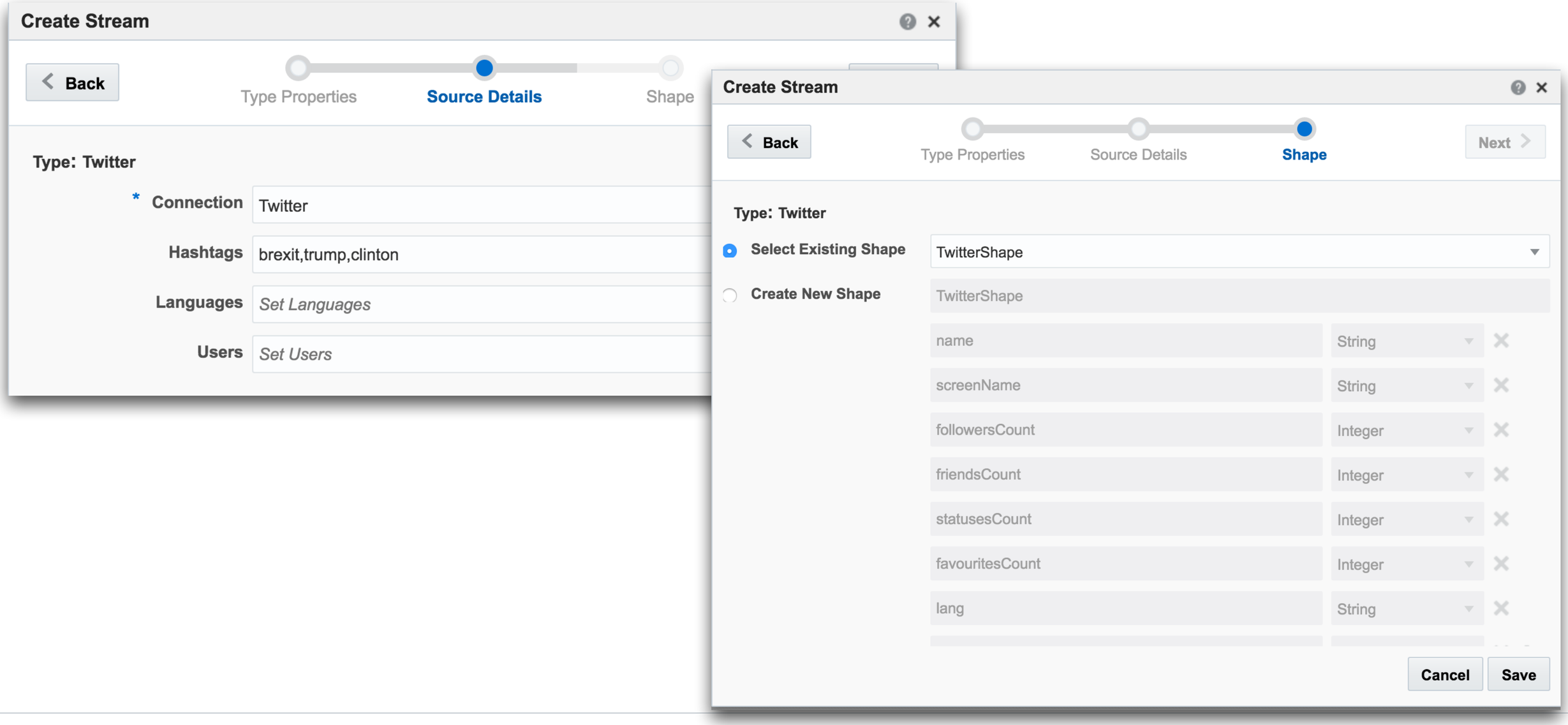
- Stream processing runtime with GUI
- Previously known as Oracle Stream Explorer
  - Part of Fusion Middleware
- Custom runtime (not WLS)
  - Future : Spark Streaming



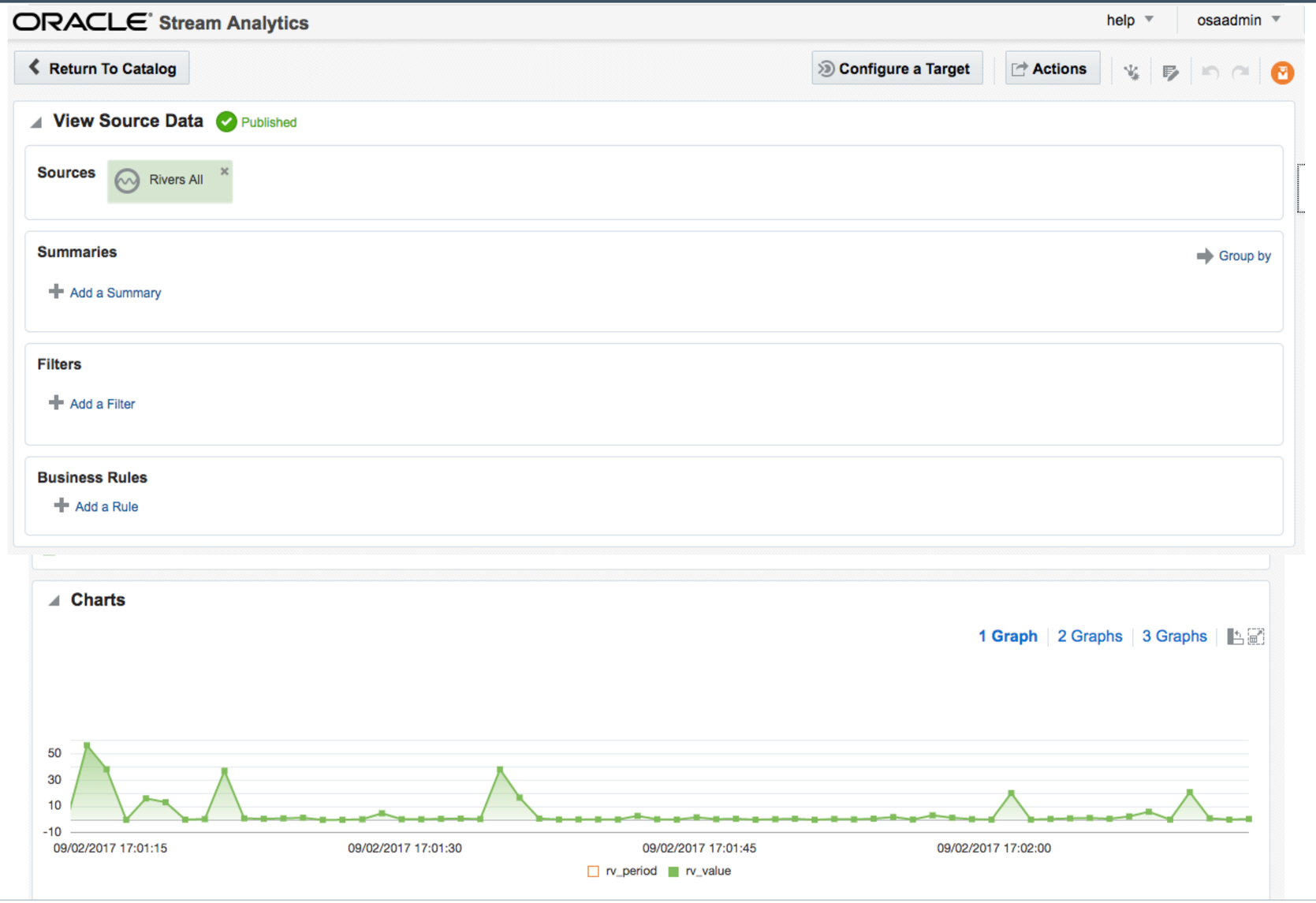
Create Connection



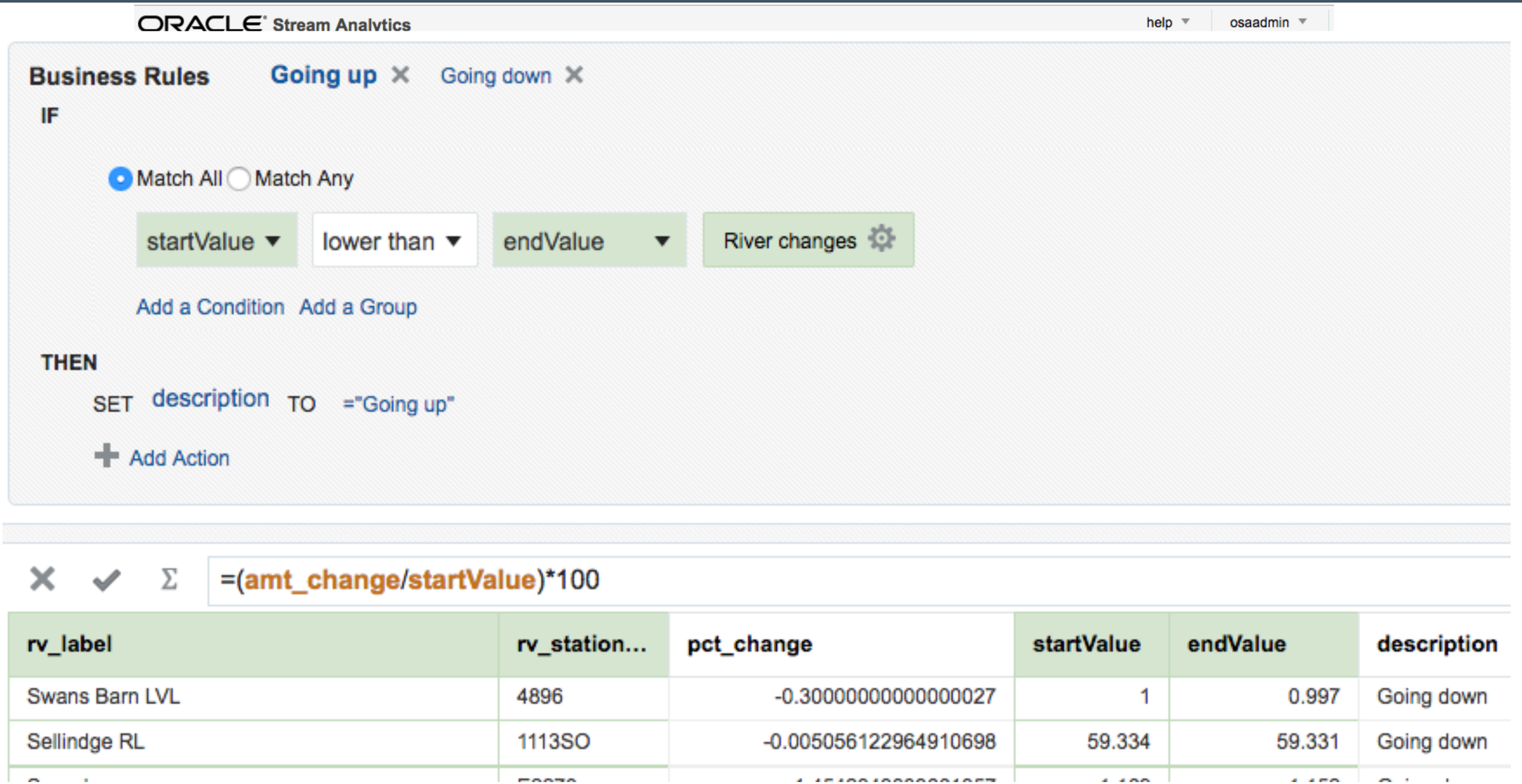
Create Stream



Exploration

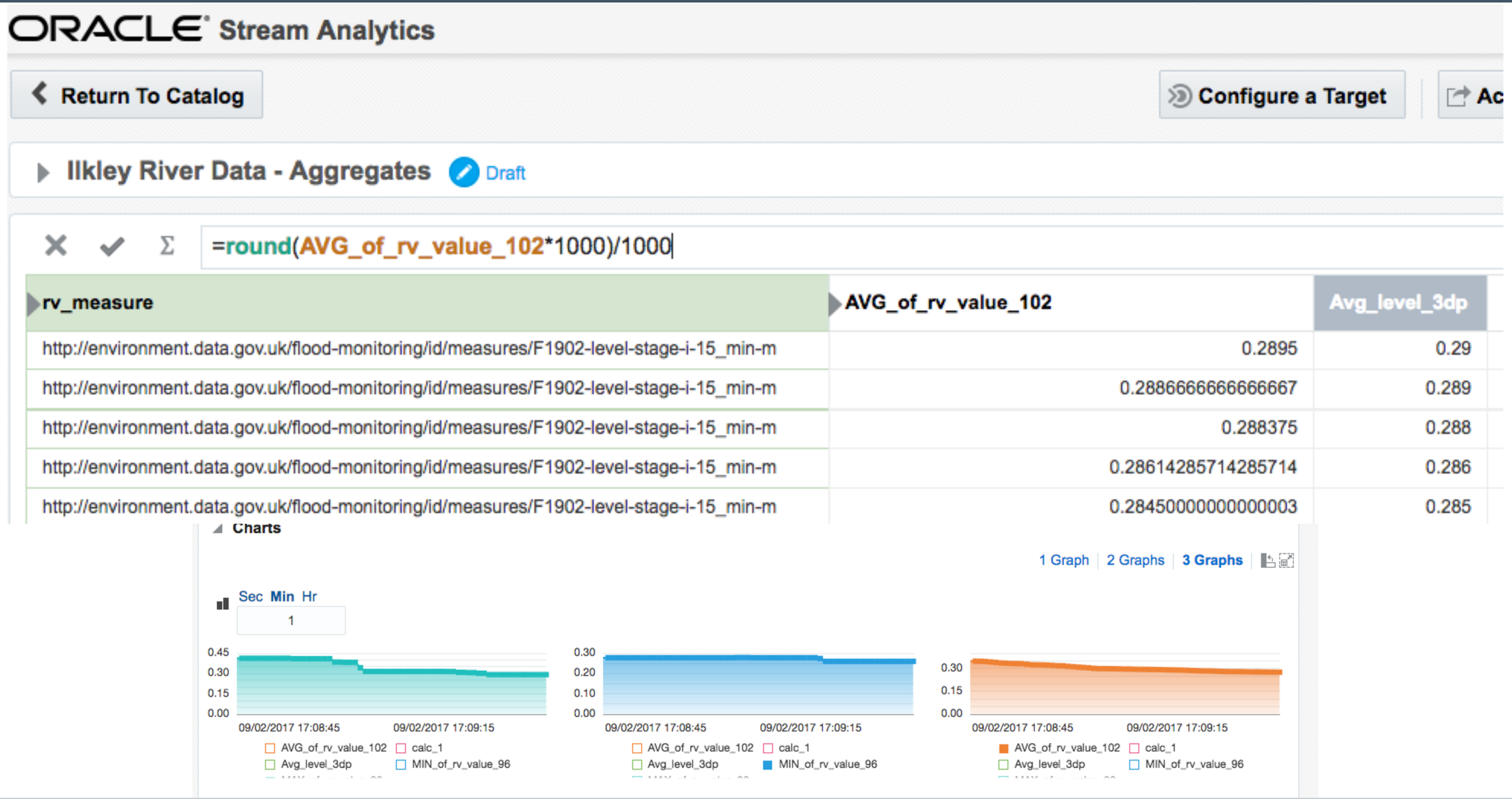


Derivations and Calculations



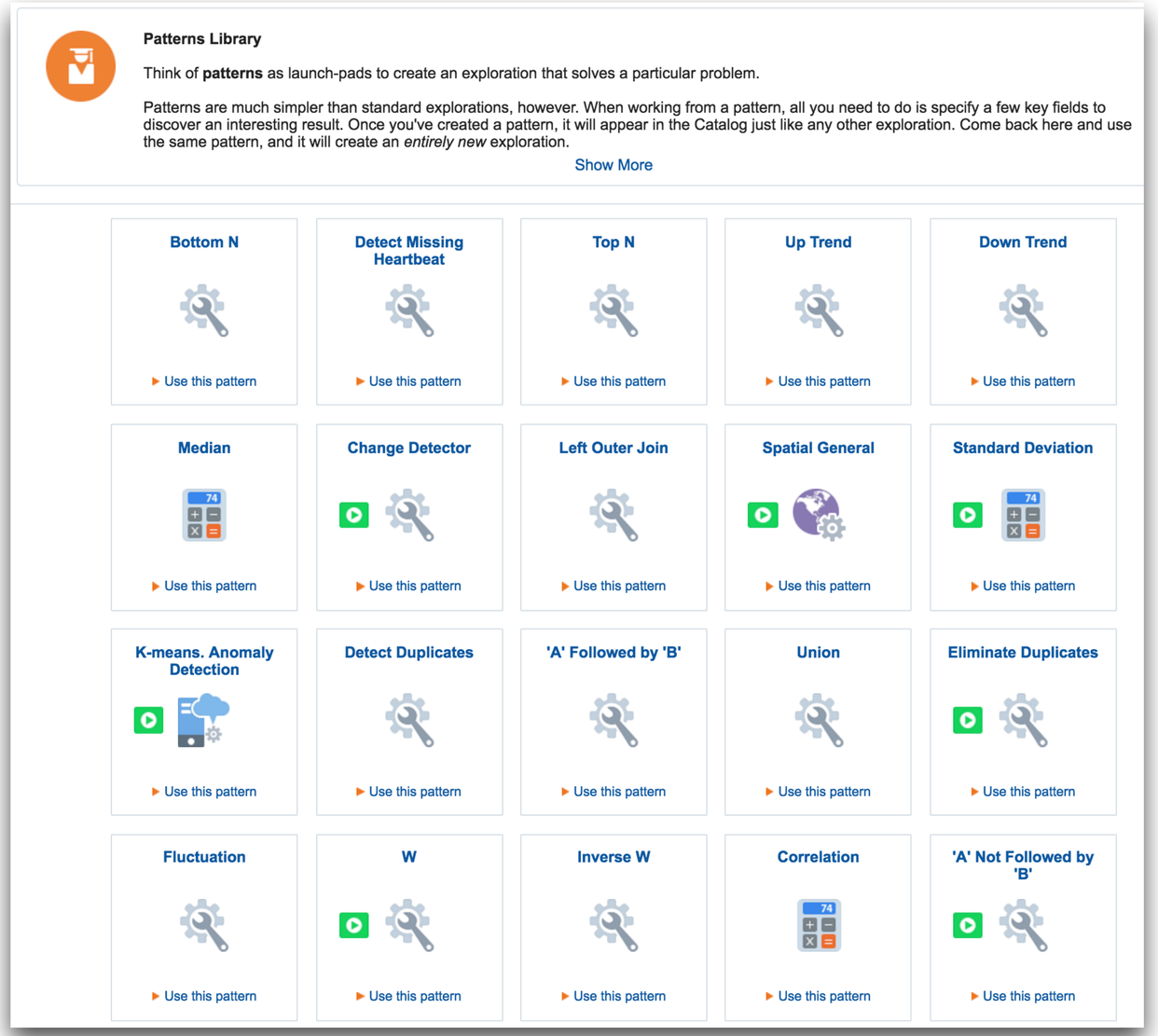
rv_label	rv_station...	pct_change	startValue	endValue	description
Swans Barn LVL	4896	-0.30000000000000027	1	0.997	Going down
Sellindge RL	1113SO	-0.005056122964910698	59.334	59.331	Going down

Aggregations



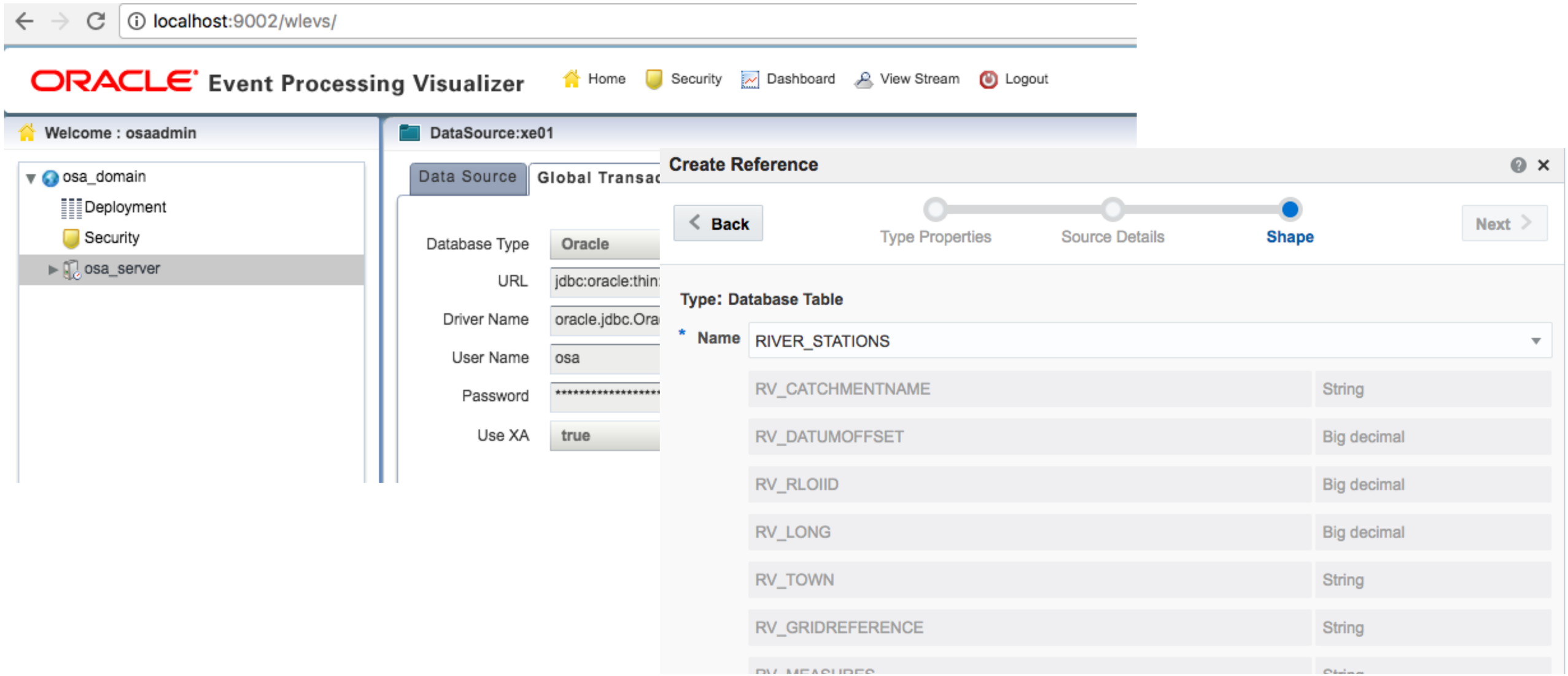
Patterns

- Pre-built templates to create Explorations
- Simple transformation and aggregation
- Built-in algorithms and spatial functionality


Patterns Library
Think of **patterns** as launch-pads to create an exploration that solves a particular problem.
Patterns are much simpler than standard explorations, however. When working from a pattern, all you need to do is specify a few key fields to discover an interesting result. Once you've created a pattern, it will appear in the Catalog just like any other exploration. Come back here and use the same pattern, and it will create an *entirely new* exploration.
Show More
Bottom N
Use this pattern
Detect Missing Heartbeat
Use this pattern
Top N
Use this pattern
Up Trend
Use this pattern
Down Trend
Use this pattern
Median
Use this pattern
Change Detector
Use this pattern
Left Outer Join
Use this pattern
Spatial General
Use this pattern
Standard Deviation
Use this pattern
K-means. Anomaly Detection
Use this pattern
Detect Duplicates
Use this pattern
'A' Followed by 'B'
Use this pattern
Union
Use this pattern
Eliminate Duplicates
Use this pattern
Fluctuation
Use this pattern
W
Use this pattern
Inverse W
Use this pattern
Correlation
Use this pattern
'A' Not Followed by 'B'
Use this pattern

Join the Streams (but never cross them…)

References


localhost:9002/wlevs/
ORACLE Event Processing Visualizer
Home
Security
Dashboard
View Stream
Logout
Welcome : osaadmin
osa_domain
Deployment
Security
osa_server
DataSource:xe01
Data Source
Global Transac
Database Type
Oracle
URL
jdbc:oracle:thin
Driver Name
oracle.jdbc.Ora
User Name
osa
Password
Use XA
true
Create Reference
Back
Type Properties
Source Details
Shape
Next
Type: Database Table
* Name
RIVER_STATIONS
RV_CATCHMENTNAME
String
RV_DATUMOFFSET
Big decimal
RV_RLOIID
Big decimal
RV_LONG
Big decimal
RV_TOWN
String
RV_GRIDREFERENCE
String

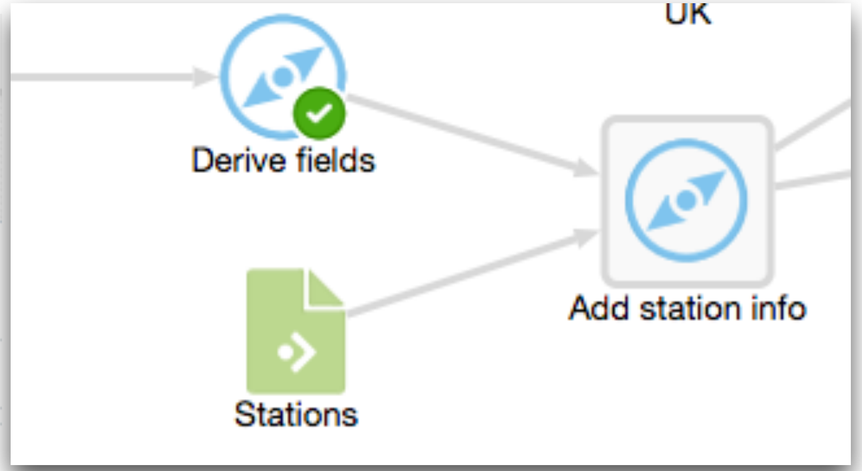
Join the Streams (but never cross them…)

ORACLE Stream Analytics

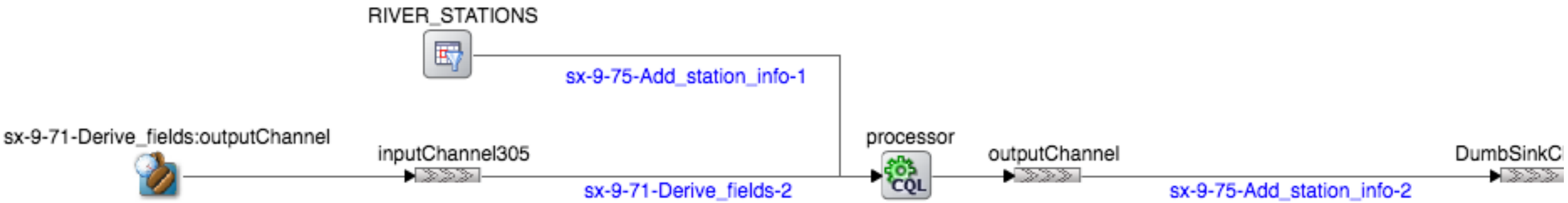
Return To Catalog

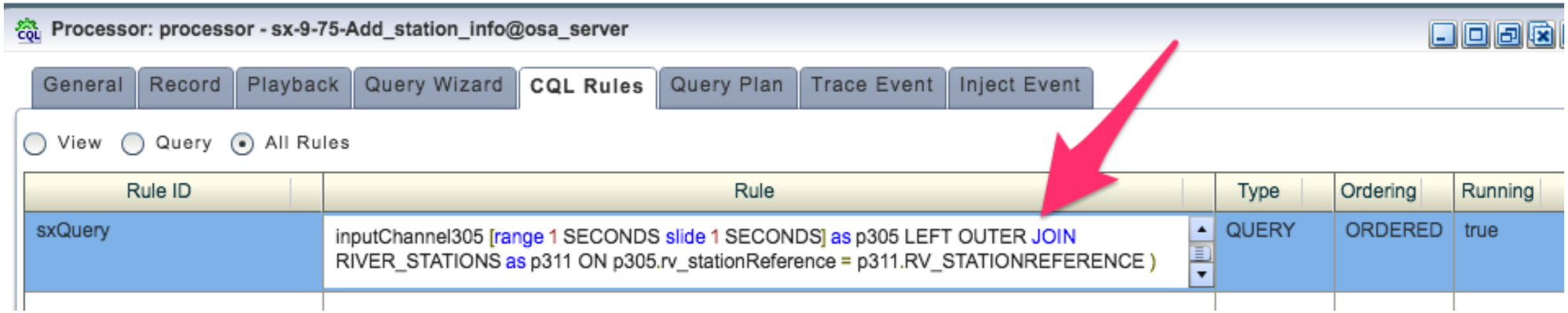
Add station info Draft

* Primary Stream	Derive fields ×
* Enriching Reference/Stream	Stations ×
* Primary Stream correlation criteria	rv_stationReference ×
* Enriching Reference/Stream correlation criteria	RV_STATIONREFERENCE ×
Window Range of the Primary Stream	1 seconds
Window Slide of the Primary Stream	1 seconds
Window Range of the Enriching Stream	1 seconds
Window Slide for the Enriching Stream	1 seconds



Under the Covers…





Join the Streams

RIVER_STATIONS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions

Sort.. | Filter:

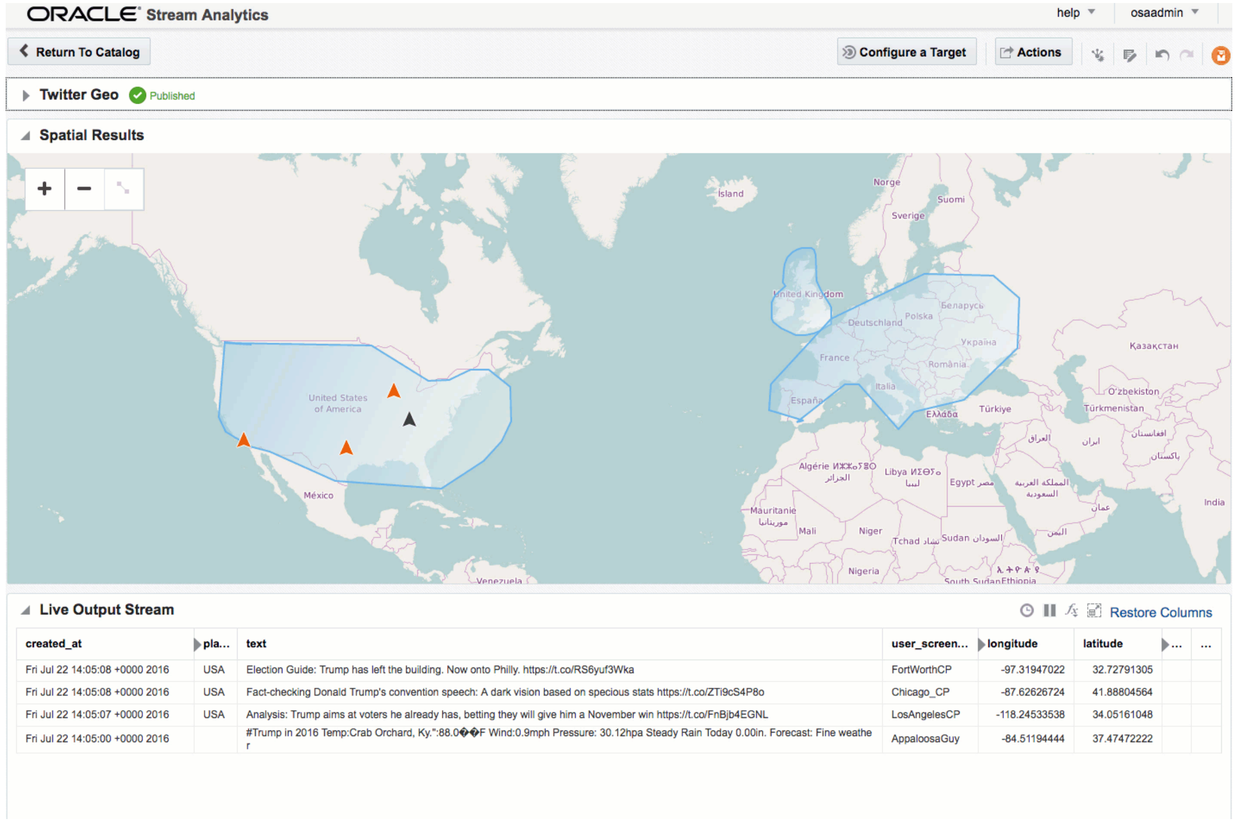
RV_STATIONREFERENCE	RV_LAT	RV_LONG	RV_TOWN	RV_RIVERNAME

rv_dateTi...	rv_st...	rv_qualif...	r...	rv_...	rv_valueT...	rv_...	RV_CA...	RV_	RV...	RV_LO...	RV_TOWN	RV_GRIDR...
2017-02-05T 04:00:00Z	531161	1	900	mAOD	instantaneous	4.486		0	0	-2.561625		ST6108975158
2017-02-05T 04:00:00Z	4041	Stage	900	mASD	instantaneous	0.319	Dove	0	2117	-1.879122	Waterhouses	
2017-02-05T 04:00:00Z	1099TH	Downstream Stage	900	mASD	instantaneous	2.567	Cotswolds	0	7046	-1.374624	Appleton	
2017-02-05T 04:00:00Z	45117	Stage	900	m	instantaneous	1.005	Exe	0	3261	-3.559651	Cowley	

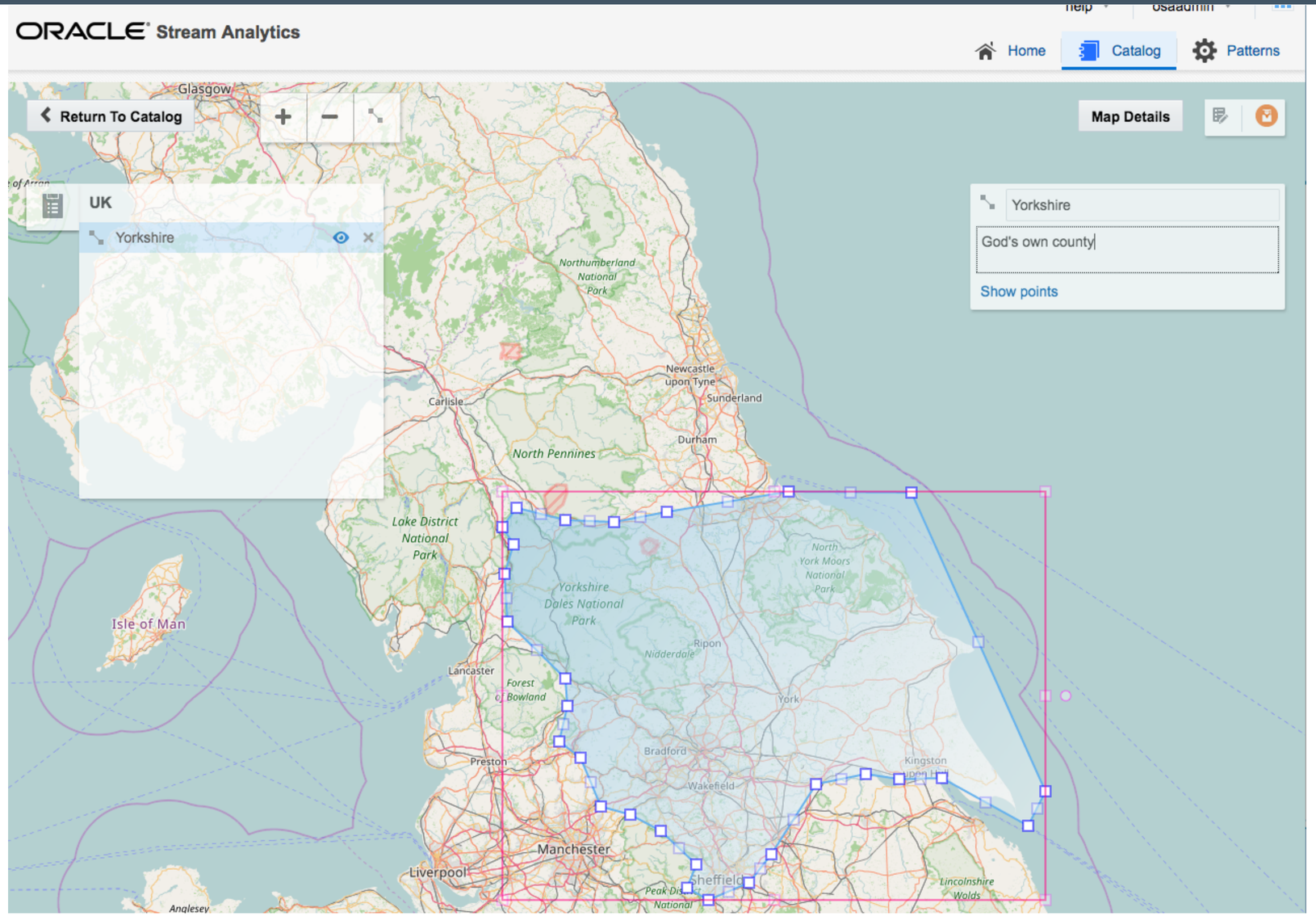
2017-02-05T 04:00:00Z	1099TH	Downstream Stage	900	mASD	instantaneous	2.567
2017-02-05T 04:00:00Z	45117	Stage	900	m	instantaneous	1.005

Spatial Pattern

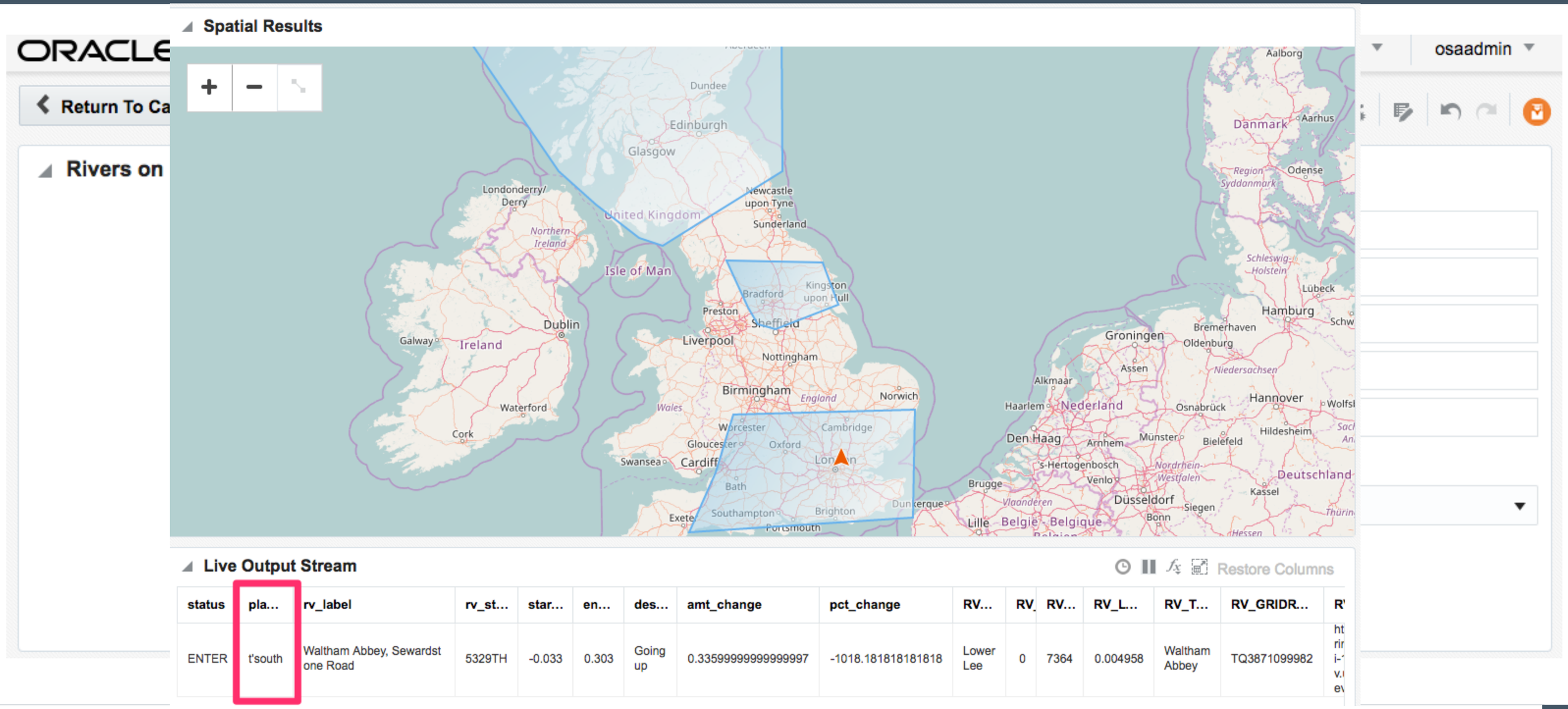
- Uses geofencing to overlay stream events and tag them as ‘entering’, ‘leaving’, or ‘staying’ in an area



Map

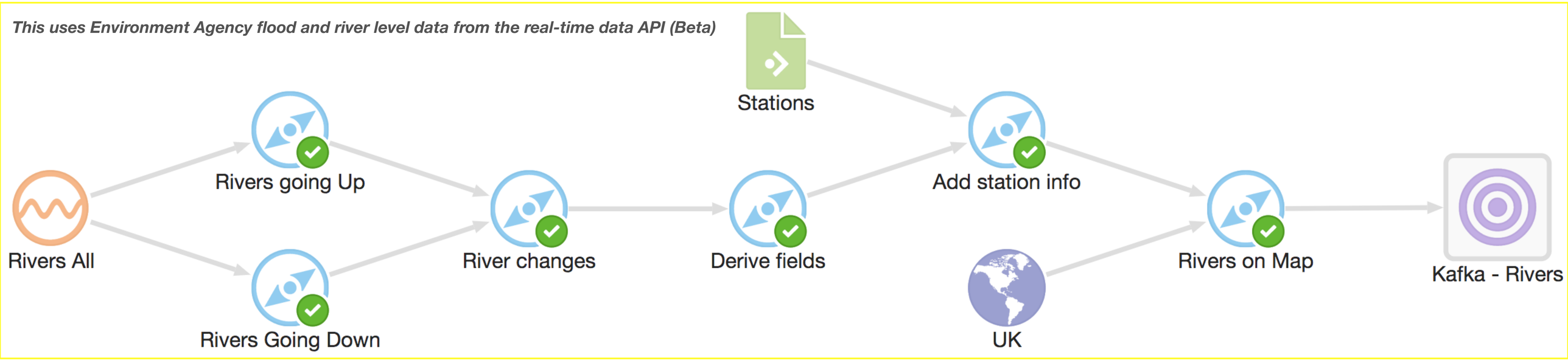

ORACLE Stream Analytics
Home
Catalog
Patterns
Return To Catalog
Map Details
UK
Yorkshire
Yorkshire
God's own county
Show points
Glasgow
Northumberland National Park
Newcastle upon Tyne
Sunderland
Carlisle
Durham
North Pennines
Lake District National Park
North York Moors National Park
Yorkshire Dales National Park
Isle of Man
Ripon
Nidderdale
Lancaster
Forest of Bowland
York
Bradford
Preston
Wakefield
Kingston upon Hull
Manchester
Liverpool
Sheffield
Lincolnshire Wolds
Anglesey

Spatial Pattern

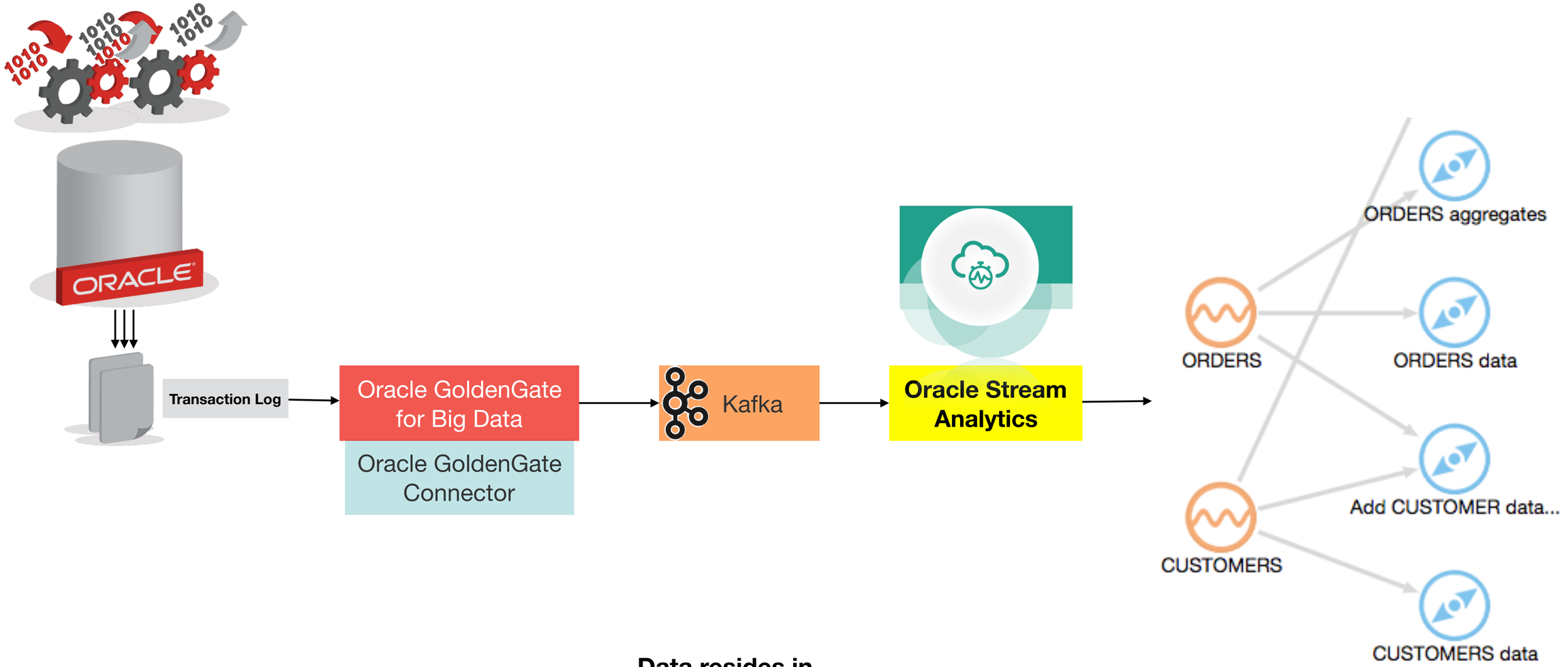


status	pla...	rv_label	rv_st...	star...	en...	des...	amt_change	pct_change	RV...	RV_	RV...	RV_L...	RV_T...	RV_GRIDR...	R'
ENTER	t'south	Waltham Abbey, Sewardst one Road	5329TH	-0.033	0.303	Going up	0.33599999999999997	-1018.181818181818	Lower Lee	0	7364	0.004958	Waltham Abbey	TQ3871099982	ht ri i- v. ev

Stream Processing Pipeline



Your Database Transactions Are Events!



Change Data Capture (CDC) streams every change made to the database into Kafka

Data resides in Kafka ready for streaming by one or more consumers

Each database table is a stream of events in Oracle Stream Analytics

Streaming from OGG to Kafka

- Oracle GoldenGate for Big Data is required
- Set up extract as before
- Replicat is defined as KafkaConnect, with JSON encoding
  - `gg.handler.confluent.type=oracle.goldengate.kafkaconnect.KafkaConnectHandler`
  - `value.converter=org.apache.kafka.connect.json.JsonConverter`
- See https://www.confluent.io/blog/streaming-data-oracle-using-oracle-goldengate-kafka-connect/

Handling Nested JSON

- OSA currently only works with “flat” JSON
- OGG nests the ‘payload’ in a sublevel of the message
  - e.g. OSA can’t currently reference `payload.LOGON_ID`


```
{
    "schema": {
        "type": "struct",
        "fields": [⟺],
        "optional": false,
        "name": "ORCL.SOE.LOGON"
    },
    "payload": {
        "table": "ORCL.SOE.LOGON",
        "op_type": "I",
        "op_ts": "2017-02-13 17:12:26.000191",
        "current_ts": "2017-02-13 17:13:29.198000",
        "pos": "00000000040000002010",
        "LOGON_ID": 42,
        "CUSTOMER_ID": 42,
        "LOGON_DATE": "2017-02-13:17:12:24"
    }
}
```

Logstash to the Rescue!

- Stream processing tool from Elastic
- Very easy to configure

- Ref:
  - http://stackoverflow.com/a/40131532/350613
  - https://github.com/logstash-plugins/logstash-filter-mutate/issues/90

```
input {
        kafka { zk_connect => 'localhost:2181'
                topic_id => 'ORCL.SOE.ORDERS'         }
}

filter {
 ruby {     code => "
        event.to_hash.delete_if {|k, v| k != 'payload'}
        event.to_hash.update(event['payload'].to_hash)
        event.to_hash.delete_if {|k, v| k == 'payload'}
     "  }
}

output {
       kafka  { topic_id => "ORCL.SOE.ORDERS_flat"
bootstrap_servers => "localhost:9092" }}
```

Flattened JSON


```
{
    "schema": {
        "type": "struct",
        "fields": [⟺],
        "optional": false,
        "name": "ORCL.SOE.LOGON"
    },
    "payload": {
        "table": "ORCL.SOE.LOGON",
        "op_type": "I",
        "op_ts": "2017-02-13 17:12:26.000191",
        "current_ts": "2017-02-13 17:13:29.198000",
        "pos": "00000000040000002010",
        "LOGON_ID": 42,
        "CUSTOMER_ID": 42,
        "LOGON_DATE": "2017-02-13:17:12:24"
    }
}
```

```
{
    "table": "ORCL.SOE.LOGON",
    "op_type": "I",
    "op_ts": "2017-02-15 17:19:12.000248",
    "current_ts": "2017-02-15 17:19:19.235023",
    "pos": "00000000040008018287",
    "LOGON_ID": 446633,
    "CUSTOMER_ID": 301132,
    "LOGON_DATE": "2017-02-15:17:19:12"
}
```

View All Data Changes As They Happen

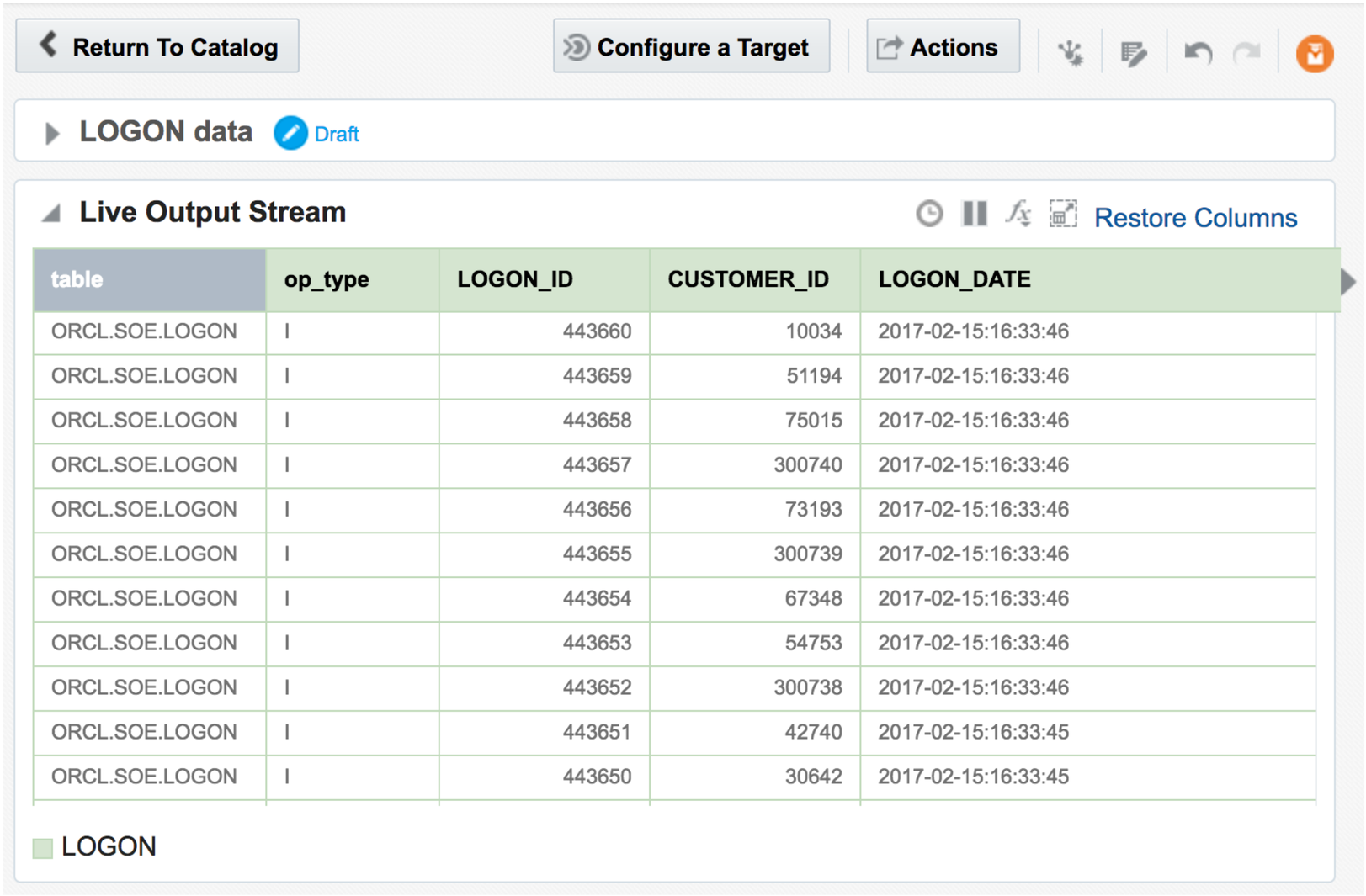
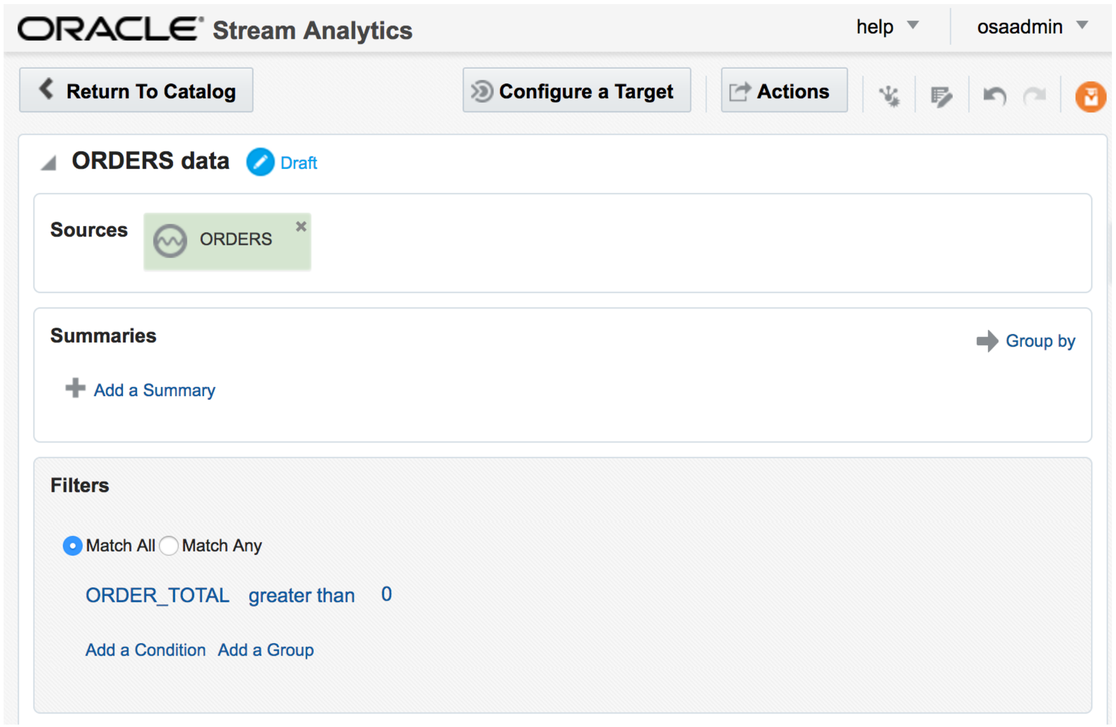
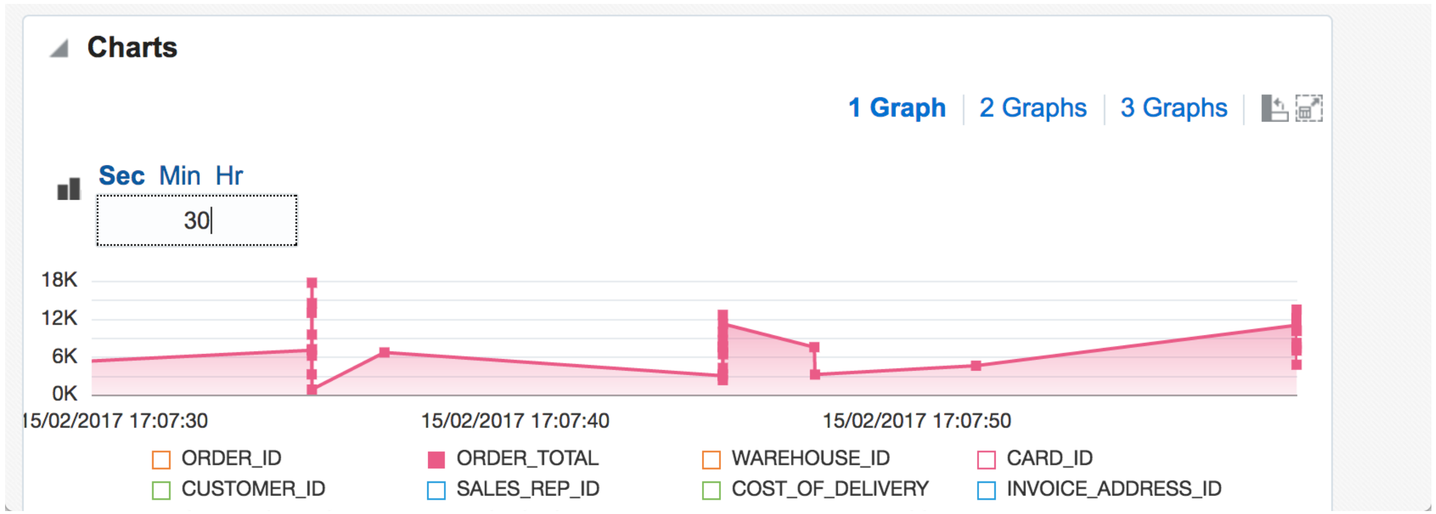


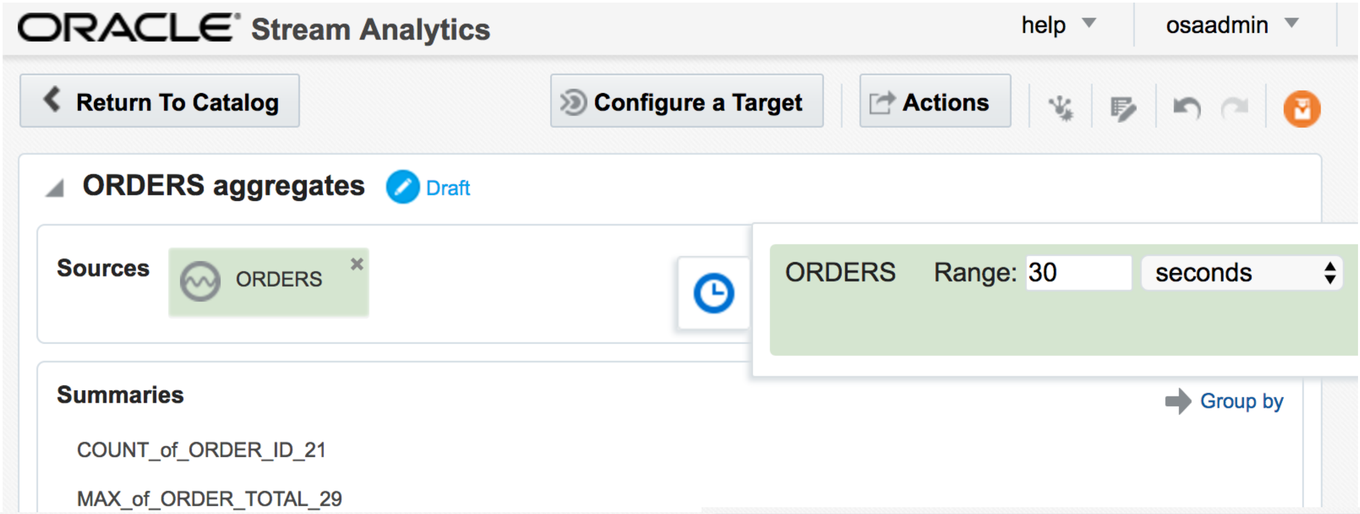
table	op_type	LOGON_ID	CUSTOMER_ID	LOGON_DATE
ORCL.SOE.LOGON	I	443660	10034	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443659	51194	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443658	75015	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443657	300740	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443656	73193	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443655	300739	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443654	67348	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443653	54753	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443652	300738	2017-02-15:16:33:46
ORCL.SOE.LOGON	I	443651	42740	2017-02-15:16:33:45
ORCL.SOE.LOGON	I	443650	30642	2017-02-15:16:33:45

Filter and Chart Stream Values





Stream Aggregates



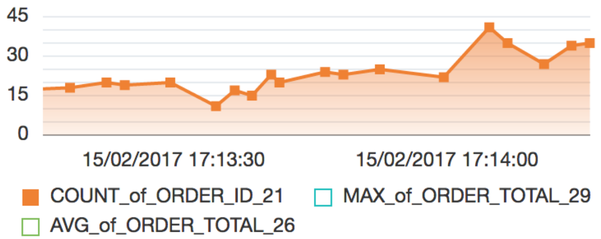
Live Output Stream

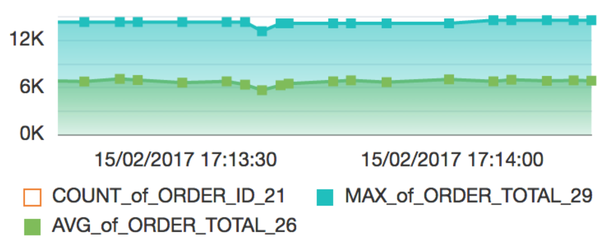
Restore Columns

order_count_30sec...	avg_order_total_30sec_roll	max_order_total_30sec_roll
22	7058.591	14173
25	6712.76	14173
23	6947.0435	14173
24	6822.5835	14173
20	6517.75	14173
23	6313.3477	14173
15	5687.4	13164
17	6386.8237	14342
11	6804.091	14342
20	6655.9	14342
19	6976.8945	14342

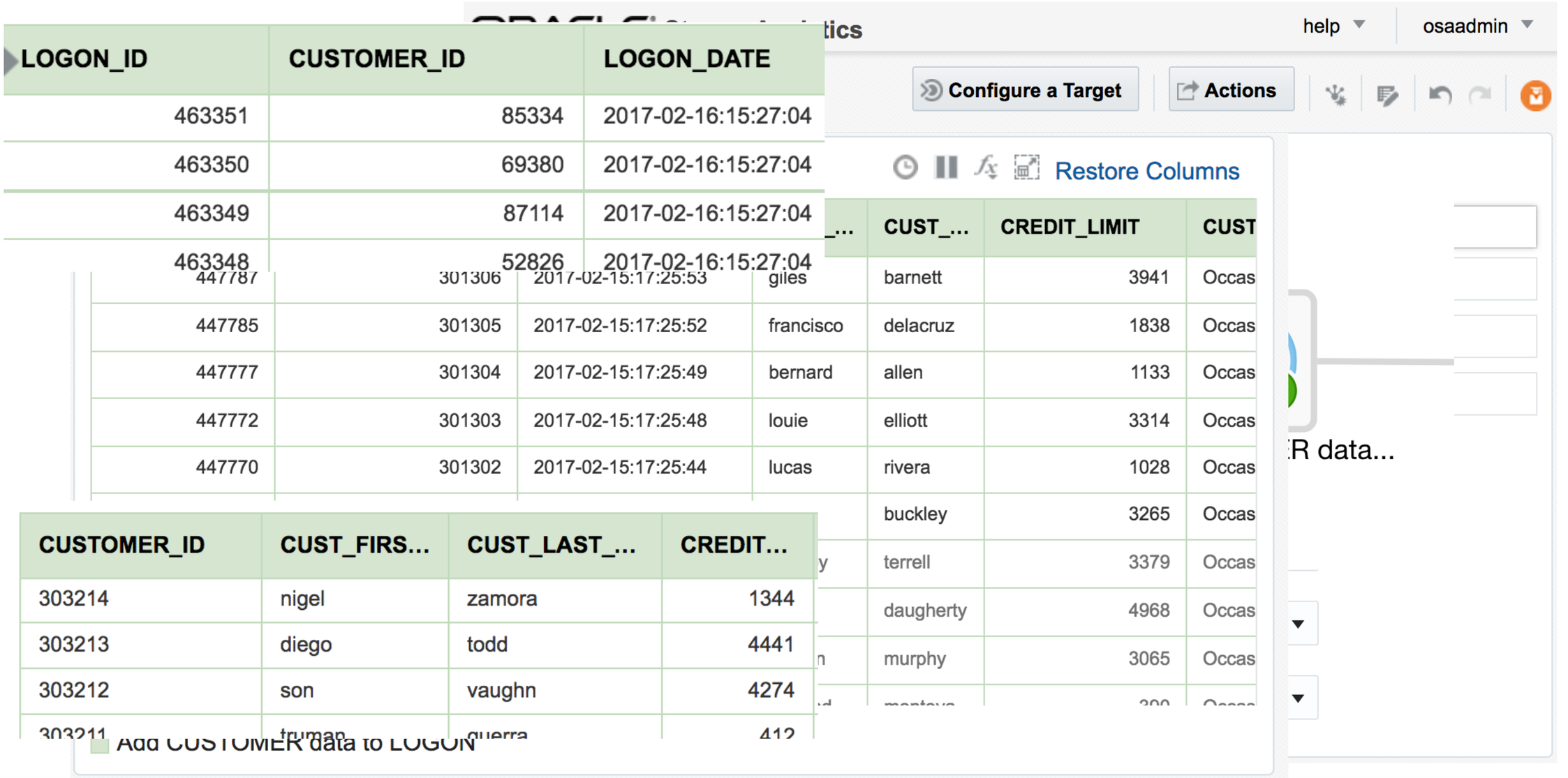
Charts

1 Graph | **2 Graphs** | 3 Graphs

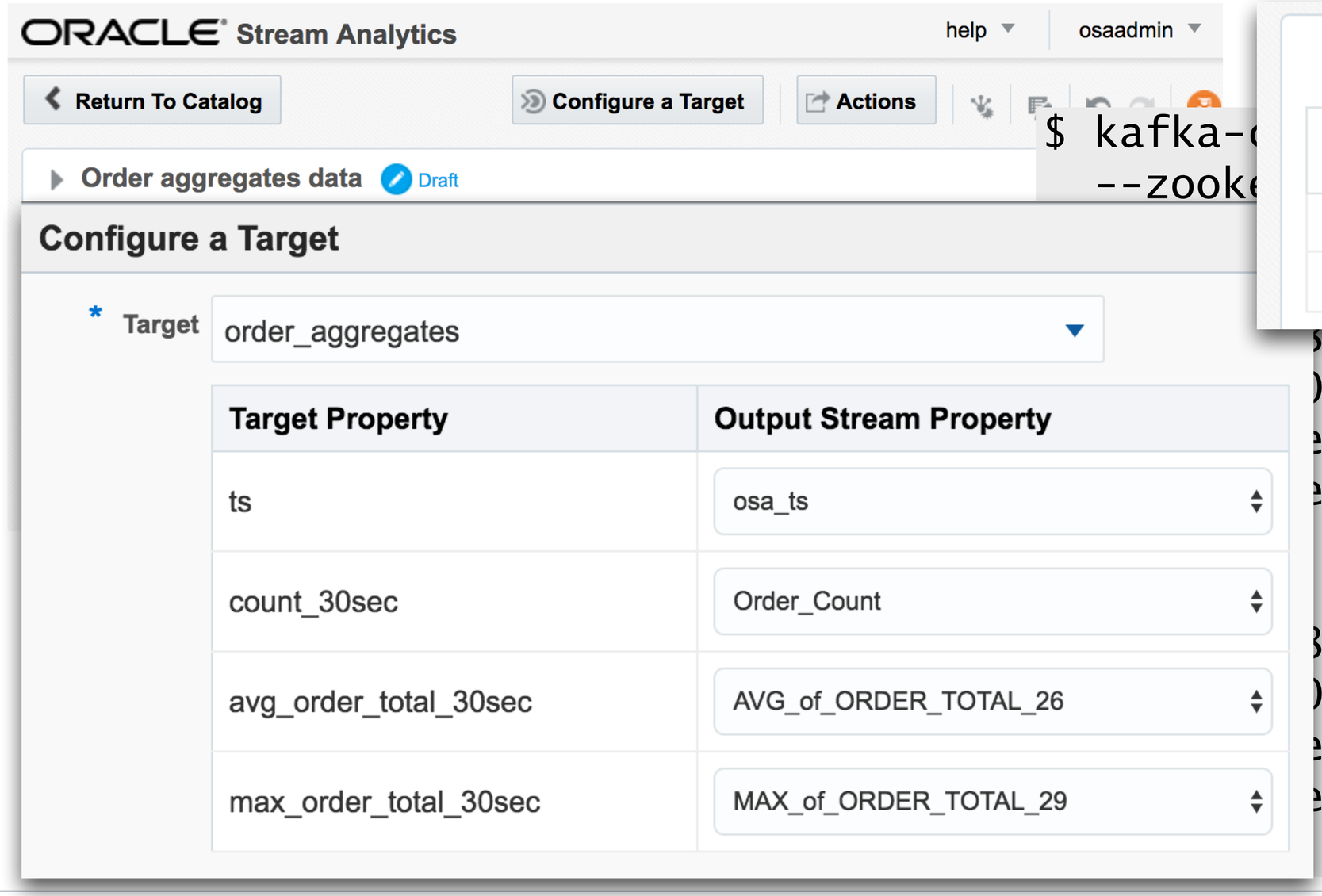


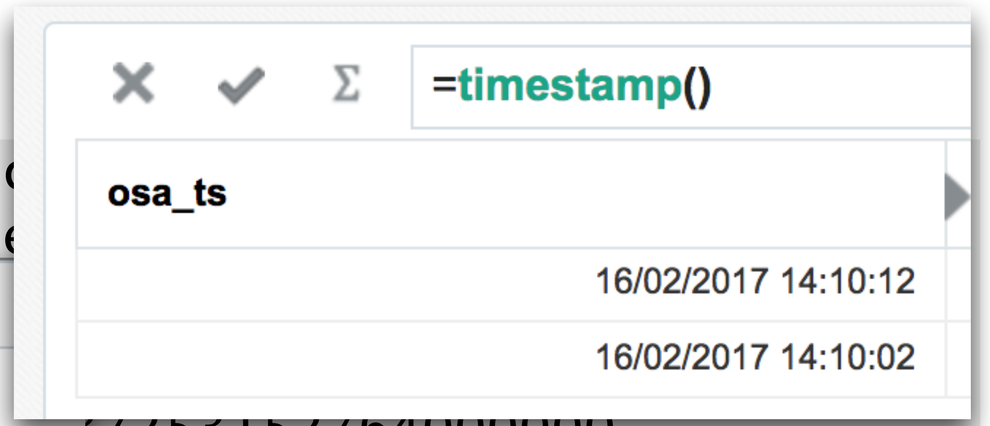


Join Streams



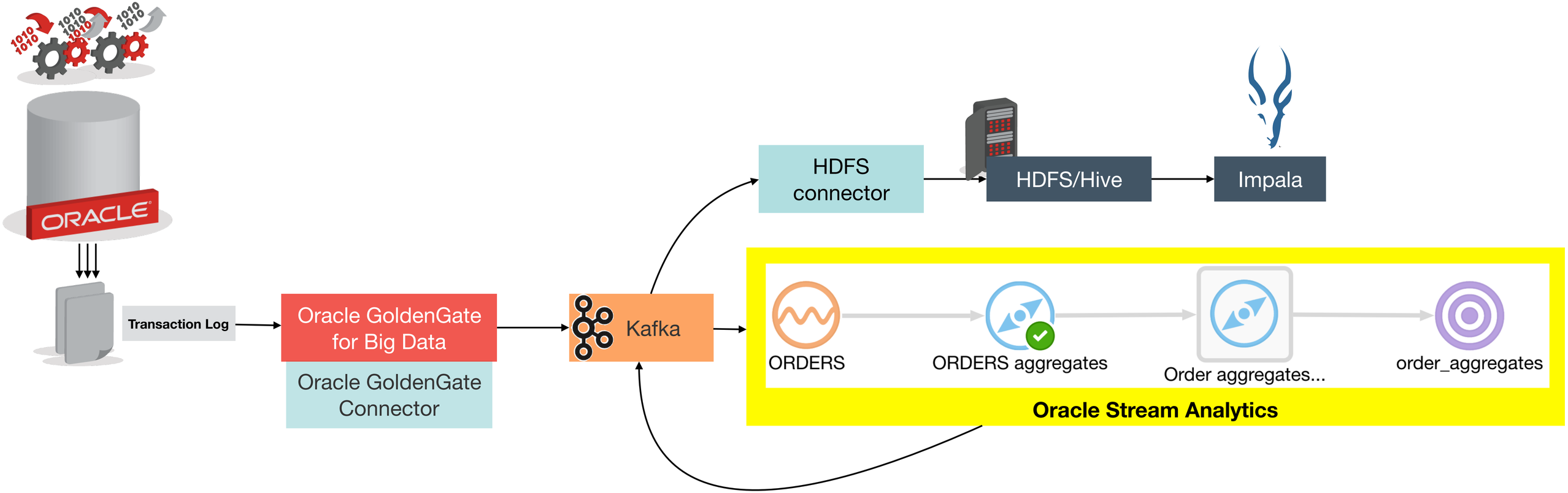
Output to Kafka






```
$ kafka-
  --zooke
                                    3725315276400000,
                                    )sec": 16,
                                    er_total_30sec": 8138.75,
                                    er_total_30sec": 15102


                                    3725316268600000,
                                    )sec": 17,
                                    er_total_30sec": 5898.5884,
                                    er_total_30sec": 15102
```

Stream Processing with Oracle Stream Analytics



Tool vs Hand-Coding

- Several powerful open-source frameworks
  - e.g. Spark Streaming, Apache Flink, Apache Kafka Streams, Google DataFlow/Apache Beam
- Generally requires specialist coders and infrastructure/ops understanding
- Parallel arguments with ETL tools vs PL/SQL

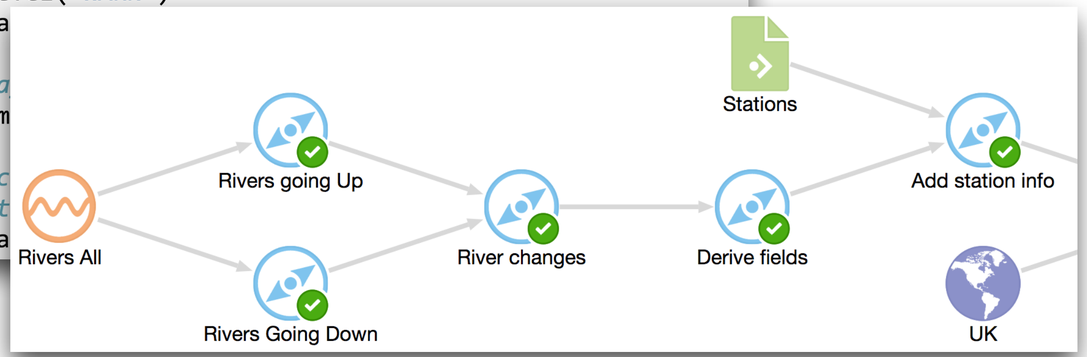
```
sc = SparkContext(appName="PythonSparkStreamingKafka_RM_02")
sc.setLogLevel("WARN")
ssc = StreamingContext(sc, 5)

# Define Kafka Consumer
kafkaStream = KafkaUtils.createStream(ssc, 'cdh57-01-node-01.m

## --- Processing
# Extract tweets
parsed = kafkaStream.map(lambda v: json.loads(v[1]))
```

Tool vs Hand-Coding

- Faster to prototype and deploy
- Accessible by non-coders
- Built-in visualisation capabilities
- Easier to support

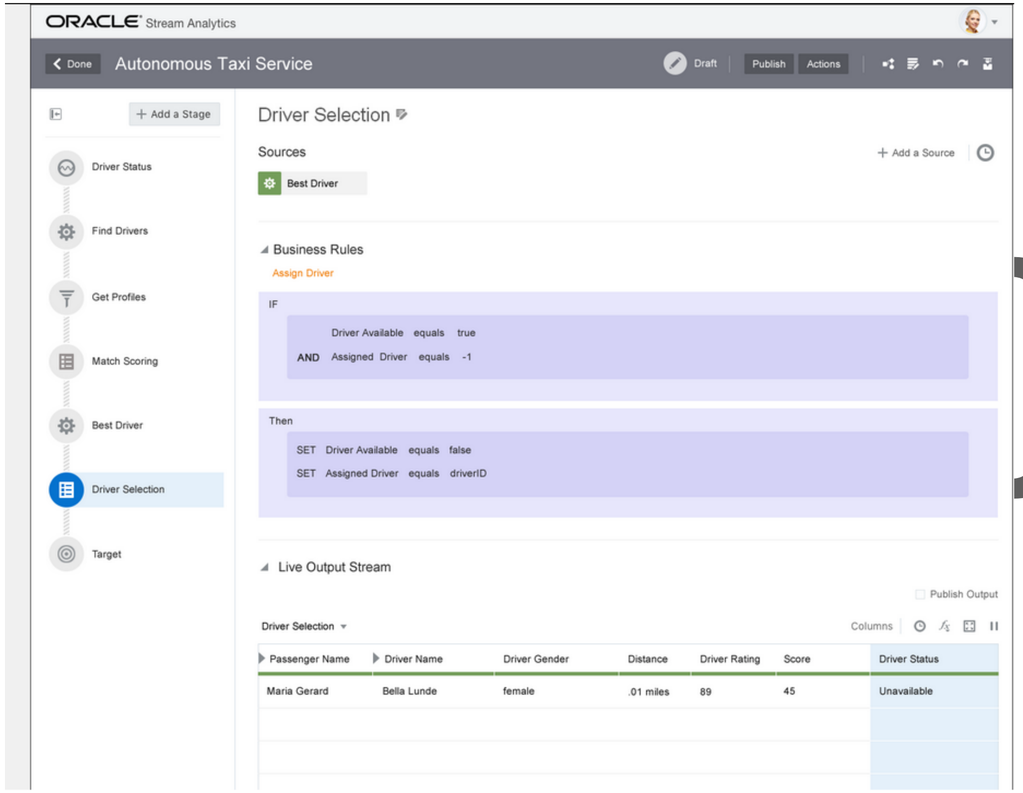
- Less flexibility
- Slower to adopt new capabilities of streaming frameworks
- License costs

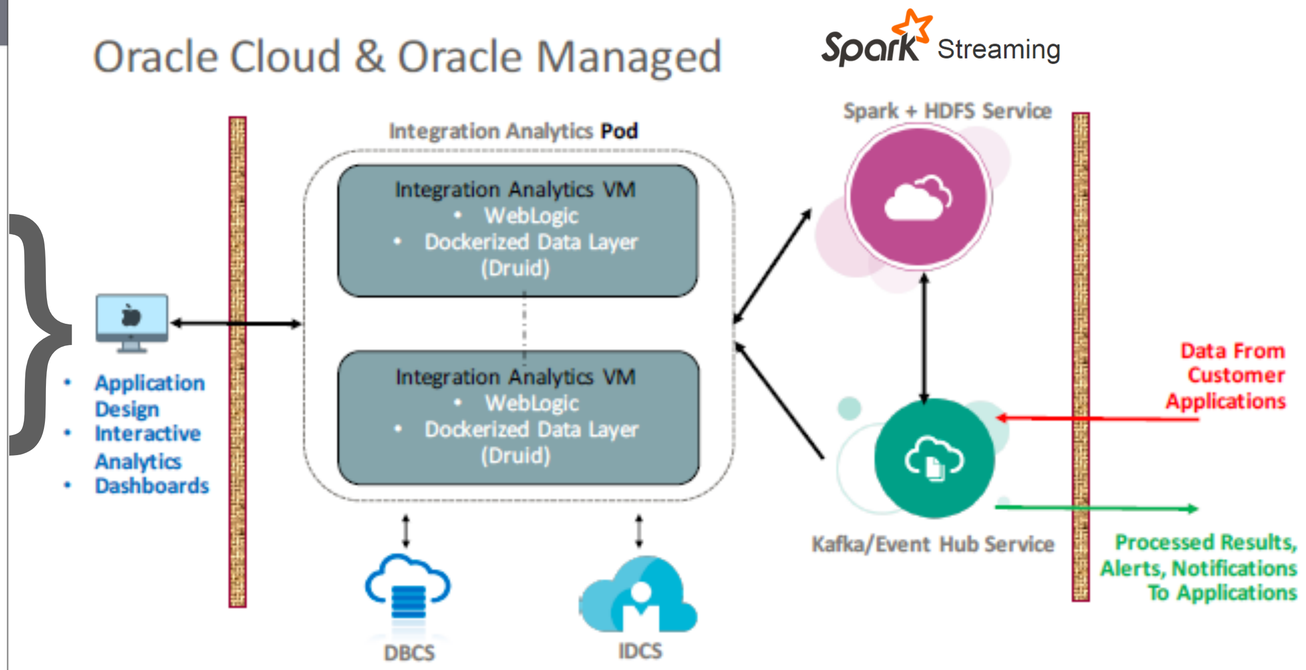
```
sc = SparkContext(appName="PythonSparkStreamingKafka_RM_02")
sc.setLogLevel("WARN")
ssc = Strea

# Define Ka
kafkaStream

## --- Proc
# Extract t
parsed = ka
```



Oracle Stream Analytics – Driving New Innovation

Fault Tolerant, Highly Available, Extreme Streaming – "with Big Data brings massive flows of Data Streams" *New Enabling Runtime Infrastructure : Spark Streaming with Kafka Messaging Layer*



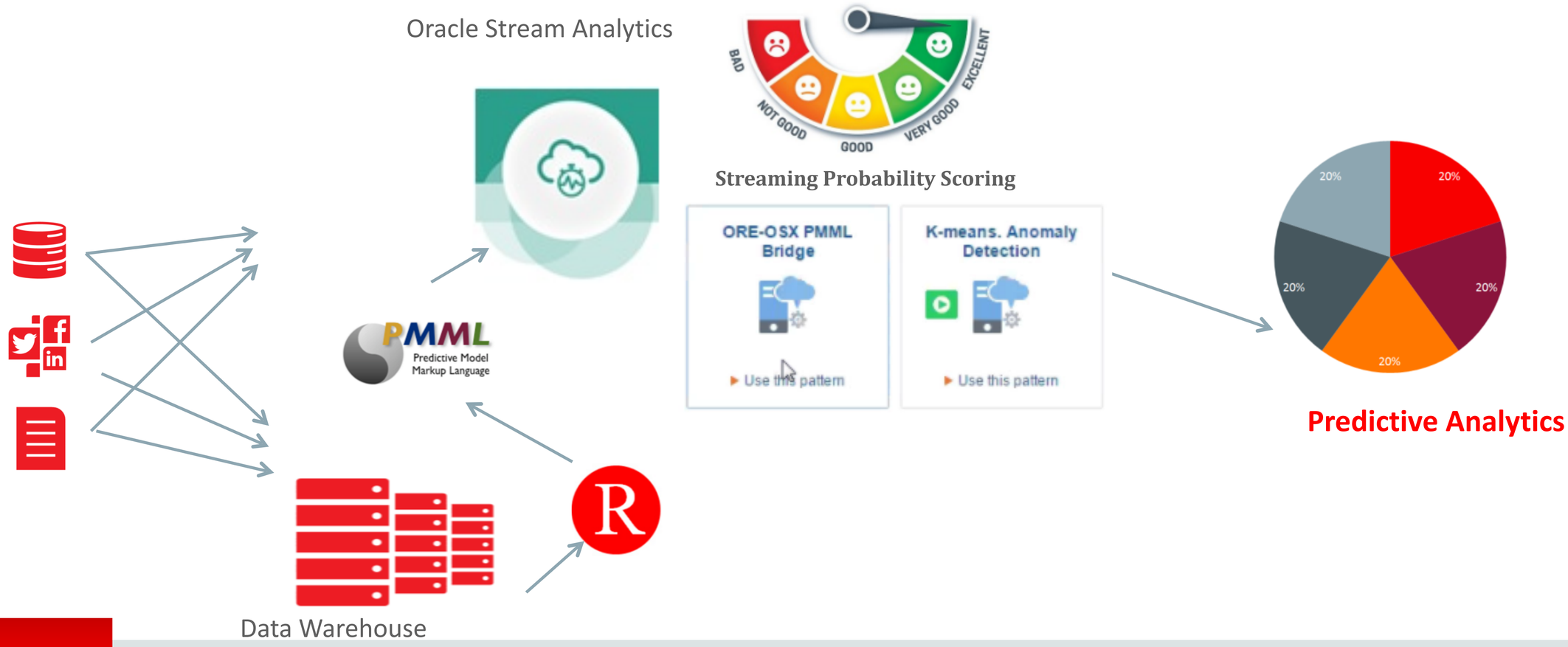


Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |

Oracle Stream Analytics – Driving New Innovation

Streaming predictive probability scoring – "I don't know what I don't know"

New PMML Pattern offering that enables OSA integration with ORE (and SparkML)



Copyright © 2016, Oracle and/or its affiliates. All rights reserved. |

#EOF

speakerdeck.com/rmoff/

email

robin@rittmanmead.com

web

http://ritt.md/rmoff

http://rmoff.net

twitter

@rmoff

irc

rmoff @ #obihackers

Reading and References

- Simple developer download/install process for OSA
- Docker image exists, enabling you to very quickly get up and running with OSA, on Mac or Windows too!
- https://www.rittmanmead.com/blog/tag/osa/
- https://www.rittmanmead.com/blog/tag/spark-streaming/